[CMPUT 466/566, Fall 2020] Machine learning

Course Project Description

Instructor: Lili Mou
LMOU@ualberta.ca
https://lili-mou.github.io/

Objectives:

1. [10 marks] The basic goal of the mini-project is for the student to gain first-hand experience in formulating a task as a machine learning problem and have a rigorous practice of applying machine learning algorithms.
2. [5 marks] The second goal (optional to undergrads) is to accomplish a non-trivial machine learning project, such as replicating a recent machine learning publication, proposing new models, and empirically analyzing machine learning models in a significant way.

   The 5 marks count as bonus for undergrads but are included within 100 total marks for grads.

Team work

Collaboration of the course project is possible only if

1) all team members have already had first-hand experience, and
2) they intend to do a non-trivial project.

The speed-dating social event is a good opportunity for students to know each other and form collaboration teams. The team has to apply to the instructor by email before Sep 18. The application may be declined if any of the team members does not have adequate machine learning background.

Proposal of a non-trivial project

A project intended to satisfy the basic requirements only (10 marks) does not need a proposal.

A non-trivial project must be follow the mandatory timeline:

- Sep 18: Notice of intent
- Oct 19: A proposal
- Dec 20: Final report

A non-trivial project requires significantly more time than a project satisfying basic requirements only, so a significant amount of time has to be set for the project.

The student must decide early if he/she is going to do a non-trivial project. The student must send a notice of intent (NOI) by Sep 18, indicating a title, a short description, and team members (optional). The NOI will not be reviewed but is mandatory for a non-trivial project.

The instructor offers a chat to anyone who sends a NOI. The chat will be scheduled by appointment initiated by the student. The instructor's availability can be found here: https://lili-mou.github.io/calendar.html

The student is supposed to read literature and prepare experimental environments after NOI. By the proposal deadline, the student must submit a pdf proposal to eClass. The instructor will read the proposal and make a comment, especially on how non-trivial the proposal is. The instructor offers another chat to those who submit the proposal.

Notice that an intended non-trivial project may not get all the 5 marks or just fall back to a basic project.

Basic Requirements [10 marks]:

- *Formulating a task into a machine learning problem*. The student CANNOT re-use any task in coding assignments (namely, house price and MNIST datasets) as the course project.
- *Implementing a training-validation-test infrastructure, with a systematic way of hyperparameter tuning*. The meaning of “training,” “validation,” “test,” and “hyperparameter” will be clear very soon.
- *Comparing at least three machine learning algorithms.* In addition, include a trivial baseline (if possible). For example, a majority guess for *k*-category classification yields *1/k* accuracy. The machine learning algorithms must be reasonable for solving the task, and differ in some way (e.g., having different hyperparameters do not count as different machine learning algorithms).

Requirements for a non-trivial project [5 marks]:

A non-trivial project could be either replicating a recent machine learning paper that involves some sophistication, proposing new models, or conducting empirically analyzing machine learning models in a significant way.

Typical, a non-trivial project involves significant amount of literature reading, programming and conducting experiments. A student would not expect any bonus mark by trying some CNN/RNN models, or applying existing code base to a new task in a straightforward way. If a student seeks non-triviality marks by replicating a recent paper, the student should assume the code base of that paper does not exist.

Final report submission:

Deadline: Dec 20, 2020 [the weekend after the final]

The submission must contain a PDF report **and** the code to reproduce the results. (Non-complying file format will result in mark deduction.)

The code can be submitted by either a zip or an online, public repo (without logging in or accepting invitation). Notice that a private GitHub repo with an invitation is not accepted.

The format of the report is flexible, but generally, the report should contain

- A short introduction, describing the background of the task
- Problem formulation (what is input, what is output, where did you get the dataset, number of samples, etc.)
- Approaches and baselines (what are the hyperparameters of each approach/baseline, how do you tune them)?

- Evaluation metric (what is the measure of success, is it the real goal of the task, or an approximation? If it's an approximation, why is it a reasonable approximation?)
- Results. (What is the result of the approaches? How is it compared with baselines? How do you interpret the results?)

Grading criteria:

Basic requirements [10 marks]:

- If the submission is not a machine learning problem, then 0 point.
- Otherwise, the grading starts from 10 points. If one or more of the above requirements are not fulfilled, it will result in mark deduction for one or a few points.
- Presentation enters the mark in a multiplicative way. The factor is 1 be default, if the report is reasonably well written.

Non-triviality [5 marks]:

- 2 marks for proposal, where 1 mark=literature review, 1 mark = proposed approach
- 3 marks based on the final report. While the project could deviate from the proposal in a reasonable way, the proposal marks may be revoked if the quality of the final project is poor.

Tips:

1. The course project only counts 10--15% of the total marks, and obviously, this course focuses more on paper derivations than experimenting. It is more important to formulate a machine learning system in a rigorous way and complete the project in time than do a super fancy project (which may require too much work and has a risk of not being finished in the course timeline).
2. Using external general-purpose machine learning packages is allowed but should be acknowledged (e.g., use libsvm to solve the task by a few lines of function call). However, using a code base directly related to your task is not allowed (e.g., download a GitHub repo and only write a few lines of script like "sh run.sh").
3. There is no constraint on the number of pages of the course report. However, the length should reflect the substance of the project, and in a normal case, a few pages suffice. An over-lengthed report will not yield a higher mark. On the contrary, it shows poor presentation skills (and may lead to mark deduction).
4. We will grade the course project in a lenient way. However, we do not accept mark negotiation for triviality judgment. Mark negotiation shows poor presentation skills, because it should have been clear in the report, thus possibly leading to mark deduction.

Collaboration and future opportunities with instructor

The instructor's research interest is mainly in natural language processing, especially focusing on unsupervised text generation and latent structure reasoning for text understanding. Students are welcome to pick one of the topics as their non-trivial course projects and are encouraged to read the related papers. If the student is indeed interested in the project, a discussion with the instructor is mandatory in this case.

The instructor is looking for MSc students as well as undergrad RAs. If a student is interested in being supervised by the instructor for future research, a course project is an ideal starting point. The instructor offers CMPUT651 (Deep Learning for Natural Language Processing) in Winter 2021 and is also willing to offer Individual Study courses in future semesters.